Relaxed Simultaneous Tomographic Reconstruction and Segmentation with Class Priors for Poisson Noise

Romanov, Mikhail; Dahl, Anders Bjorholm; Dong, Yiqiu; Hansen, Per Christian

Publication date:
2015

Document Version
Publisher's PDF, also known as Version of record

Relaxed Simultaneous Tomographic Reconstruction and Segmentation with Class Priors for Poisson Noise

Mikhail Romanov, Anders Bjorholm Dahl, Yiqiu Dong and
Per Christian Hansen

November 4, 2015

Abstract

This work is a continuation of work on algorithms for simultaneous reconstruction and segmentation. In our previous work we developed an algorithm for data with Gaussian noise, and in that algorithm the coefficient matrix for the system is explicitly store. We improve this algorithm in two ways: our new algorithm can handle Poisson noise in the data, and it can solve much larger problems since it does not store the matrix. We formulate this algorithm and test it on artificial test problems. Our results show that the algorithm performs well, and that we are able to produce reconstructions and segmentations with small errors.

1 Introduction

In this work we continue developing the method of Simultaneous Reconstruction and Segmentation (SRS) that makes reconstruction and segmentation in joint fashion that first was presented in the work [**?**] and was further developed in [**?**]. The main goal of this work is to solve larger problems and to apply the algorithm to problems with Poisson noise. We will call our method in this work SRS-II as it is different from the one we presented in the first work. To make the search for a minimum of a non-convex function more robust, we use approach similar to the Simulated Annealing approach. We test our approach on an artificial problem.

There is a variety of methods that allow to make reconstruction. The main analytical method is Filtered Back Projection (FBP), also known as inverse Radon Transform [**?**], [**?**]. The strong sides of this algorithm are simplicity and low computational time, but many projections are usually needed for a good reconstruction. In case the amount of data that is provided for the reconstruction is not enough, the results will be poor. Another drawback of this algorithm is that it is significantly affected by noise. In case the data that was collected is not very precise - the reconstruction may be noisy. This drawback in many cases may be compensated by amount of data, but collecting big amounts of data in many cases is unwanted or expensive. One more drawback of this approach is that the supported geometries are very limited.

Another class of important techniques are the algebraic reconstruction techniques [**?**]. This includes Kaczmarz method [**?**], Cimmino method [**?**] and many others. These methods rely on the phenomenon of semiconvergence [**?**]. The main advantage of these techniques is that they support any reconstruction geometry as long as it may be formulated in the form of a set of linear system of equations $\mathbf{b} = A\mathbf{x}$, where A is a matrix, $\mathbf{x}$ is a vector that represents object, $\mathbf{b}$ is the vector that represents the measurements. These methods are easy to use and well researched. There are several drawbacks of these algorithms. One of them is that these methods usually do not take into account the specific types of the noise that may appear during the measurement process. Another drawback is that these methods usually are not noise robust: small deviations in the data may lead to big deviations in solutions in case the problem is underdetermined.

Next, quite important set of methods are variational techniques [**?**], [**?**]. These techniques are based on well-known and widely used optimization algorithms. The main idea of this approach is to formulate the reconstruction problem as an optimization problem and after that to solve this problem with any of available optimization technique. To solve the optimization problem fast and precise, usually, the problem should be convex and the function under optimization should have an analytical gradient. This approach is significantly more flexible than algebraic reconstruction techniques. As the problem may be solved as soon

as it is convex, the formulated problem may be quite complex. The method that is described below belongs to this class of the approach.

One of the most interesting variation of this approach for us is the Total Variation (TV) reconstruction method [**?**]. It's main idea is to maximize the likelihood of observing the data that was received during the measurements (that in the future we will call data fitting term) and, in the same time, to minimize the integral of the norm of the gradient in the image (this is usually referred to as a regularization term). The advantages of this approach are: simplicity, usage of well researched optimization techniques, predictable behaviour, noise robustness. Also, to make a good reconstruction with this method less data is needed than with all the methods that were described above. One of the main aims of this method is to prevent smoothening of the edges. As for the disadvantages, this methods in many cases smears out some important small details. Also, the staircasing is a well-known artifact of this method (usually happens on the edges of the object or on the regions with a gradient, substitutes the gradient or the edge with a set of steps with different intensity that is the same inside of the step) [**?**].

The Monte-Carlo approaches to these problems are able to find the global minimum of the objective function. It solves non-convex problems, but the long time of computing the solution is a significant drawback of this approach. Because of that this is very rarely used to solve real problems. As an example of a good problem for this approach is a reconstruction with l_0 norm of the gradient of the image that corresponds to penalty of the length of the edges in the image. We do not consider this approach as a good candidate to solve our problem.

As well as for the reconstruction, for segmentation there exist many approaches to do a segmentation. One of the main direction here is called snakes [**?**]. This approach needs manual input and thus is not good for our application as we want to register the regions automatically without any interaction with human.

Another approach in this area is based on Markov Random Fields (MRF) such as Potts model, where for each of the pixels (or in some applications, regions) one label is assigned as a class to which the pixel belongs. Usually in this approach the algorithm that is used to get the results is Graph-Cut method [**?**], [**?**], [**?**]. In case of more than two labels classification, Alpha-Expansion algorithm (modification of Graph-Cut method) is used, although this approach gives only the approximation to the solution as overall problem in case of non-binary segmentation is NP-hard as the problem is discrete.

Finally, another popular set of segmentation methods is called Level-Set methods[**?**], [**?**], [**?**]. These methods are real-valued methods that use convex optimization techniques to find the segmentation. The advantage of these methods is that in case of well-formulated (i.e., convex and has an analytical gradient) problem, it will output the only minimum with given precision. Also, for the majority of the optimization techniques, the dependency of computational time on precision is well-known.

In our work as a model for a segmentation we use a Hidden Markov Measure Field Models (HMMFM) [**?**]. The main idea of the HMMFM is to assign to each of the pixels the probability with which it belongs to each of the classes. Thus, the HMMFM may be considered as a relaxation of the MRF. The problem of computing the optimal HMMFM may be easily formulated as an optimization problem with a regularization term and may be easily solved as soon as the problem is convex.

Usually in the applications one does reconstruction first and after that makes a segmentation of the received segmentation. This approach itself usually gives good results, but in case the expected classes are known, does not utilize all the available information to generate a good reconstruction output. More than that, in case some errors occur on the reconstruction step - they are likely to propagate to the segmentation of the reconstruction.

We would like to improve the results of both reconstruction and segmentation by utilizing information about the classes of materials that are likely to exist in the object and by incorporating the segmentation and the reconstruction into one joint procedure.

To accomplish this we use variational approach for both reconstruction and segmentation problems. As a model for segmentation we use Hidden Markov Measure Field Models (HMMFM). This makes our problem a constrained optimization problem. We use a statistical approximation to make this problem convex. After that we are able to apply standard optimization techniques to compute reconstruction and segmentation. For the image optimization we use well-known L-BFGS algorithm [**?**] and for HMMFM optimization we use Frank-Wolfe algorithm [**?**].

Table 1: A simple example of the HMMFM; the numbers in the table are the probabilities in the HMMFM for each class and each pixel.

class k	pixel 1	pixel 2	pixel 3	pixel 4	$\cdots$	pixel j	$\cdots$	pixel N
1	0.1	0.2	0.2	0.3	$\cdots$	δ_{j1}	$\cdots$	0.9
2	0.0	0.1	0.2	0.0	$\cdots$	δ_{j2}	$\cdots$	0.0
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$		$\vdots$		$\vdots$
K	0.8	0.6	0.3	0.0	$\cdots$	δ_{jk}	$\cdots$	0.1
$\sum_{k=1}^{K} \delta_{jk}$	1.0	1.0	1.0	1.0	$\cdots$	1.0	$\cdots$	1.0

2 Problem formulation

We solve the problem of reconstruction of an image $\mathbf{x}$ given the set of projections $\mathbf{b}$. We denote the values of the pixel j of the image as x_j and the recorded projection i value as b_i. The projection matrix is denoted as A in the matrix form and the element that encodes the length of the ray i inside of the pixel j is denoted as a_{ij}. The projections $\mathbf{b}$ and the projection matrix A are known. Thus, the forward model can be formulated as

$$A\mathbf{x} = \mathbf{b}. \tag{1}$$

We consider the noise to be Poisson noise.

We start with the summary of the theory from our previous work and then develop it for our new problem.

We assume the classes of materials that could be found in the reconstructed image known. We consider that the classes have Gaussian distribution of the attenuation coefficients with the mean values μ_k and standard deviations σ_k, where k is the index of the class.

To do the segmentation of the image we use the concept of the Hidden Markov Measure Field Model (HMMFM). We denote the HMMFM as matrix δ. The HMMFM can be represented as a table of probabilities of the pixel to belong to the specific class. The HMMFM illustration can be found on the Table 1. The value of the HMMFM for the pixel j and class k we denote as δ_{jk}. Thus, the HMMFM over the classes should sum up to one in each pixel:

$$\begin{aligned} \forall\ j\ \sum_k \delta_{jk} = 1; \\ \forall\ j,k\ \delta_{j,k} \geq 0. \end{aligned} \tag{2}$$

It is easy to formulate the problem of reconstructing $\mathbf{x}$ and δ from $\mathbf{b}$ in terms of probability maximization:

$$\begin{aligned} \mathbf{x}^*, \delta^* = &\arg\max_{\mathbf{x},\delta} p(\mathbf{x}, \delta|\mathbf{b}) \\ \text{s.t.} \quad &\forall\ j\ \sum_k \delta_{jk} = 1; \\ &\forall\ j,k\ \delta_{j,k} \geq 0 \\ &\forall j \quad x_j > 0 \end{aligned} \tag{3}$$

This problem can be reformulated using Bayes rule:

$$p(\mathbf{x}, \delta|\mathbf{b}) = \frac{p(\mathbf{b}|\mathbf{x}, \delta)p(\mathbf{x}|\delta)p(\delta)}{p(\mathbf{b})}. \tag{4}$$

Here the probability of the measured data given the HMMFM δ and the image $\mathbf{x}$ can be written as $p(\mathbf{b}|\mathbf{x})$ because the data $\mathbf{b}$ depends only on the image $\mathbf{x}$. Due to the positivity of the probability and the monotonicity of the logarithm, the problem (3) can be represented as a maximization problem of the logarithm of the probability $p(\mathbf{x}, \delta|\mathbf{b})$:

$$\log p(\mathbf{x}, \delta|\mathbf{b}) = \log p(\mathbf{b}|\mathbf{x}) + \log p(\mathbf{x}|\delta) + \log p(\delta) - \log p(\mathbf{b}). \tag{5}$$

The probability $p(\mathbf{b})$ does not depend on variables under optimization ($\mathbf{x}$ and δ). Therefore, the optimization problem without the last term and with it have the same optimal point meaning that we can consider the problem without it:

$$\log p(\mathbf{x}, \delta|\mathbf{b}) = \log p(\mathbf{b}|\mathbf{x}) + \log p(\mathbf{x}|\delta) + \log p(\delta) + \text{const}. \tag{6}$$

In this formula the first term is a data fidelity term, the goal of this term is to fit the data the best possible way. In this work we consider only the problems, where the projections are obtained via measurements of the photons. In this problem the measured values are distributed according to the Poisson distribution:

$$p(\mathbf{b}|\mathbf{x}) = \prod_i \frac{\lambda_i^{b_i} \exp(-\lambda_i)}{b_i!}, \tag{7}$$

where the λ_i are the expected value of the measured values on the detector i. In case of the Emission Tomography the Poisson distribution has the following expected values for the measurements:

$$\lambda_i = \sum_j a_{ij} x_j. \tag{8}$$

In our approach we maximize the logarithm of the probability instead of the probability itself. Taking logarithm of the equation (7), we get the following expression taking into account that $\log(b_i!)$ does not depend on the values λ_i and, thus, does not depend on $\mathbf{x}$ due to (8):

$$\log p(\mathbf{b}|\mathbf{x}) = \sum_j (b_i \log \lambda_i - \lambda_i) + \text{const}. \tag{9}$$

The second term $\log p(\mathbf{x}|\delta)$ in (6) is a class fidelity term. The probability under the logarithm is the product of the weighted mixture of Gaussians for each pixel, where weights are defined by the HMMFM, while the parameters σ_k, μ_k of the Gaussians are assumed to be known:

$$p(\mathbf{x}|\delta) = \prod_j \sum_k \delta_{jk} \frac{1}{\sqrt{2\pi\sigma_k}} \exp\left(-\frac{(x_j - \mu_k)}{2\sigma_k^2}\right). \tag{10}$$

The third term $\log p(\delta)$ of the sum in (6) is the statistical prior for the HMMFM. In this work we use the neighbourhood prior: in case a pixel is more likely to belong to a specific class, then the neighbours are encouraged to belong to the same class:

$$p(\delta) = \exp\left(-\Phi(\delta)\right). \tag{11}$$

Here the function $\Phi(\delta)$ is an l_2 norm of discrete approximation of the gradient for each of classes of the HMMFM:

$$\Phi(\delta) = \sum_{j,k} \sum_{j' \in N(j)} (\delta_{j,k} - \delta_{j',k})^2, \tag{12}$$

where $N(j)$ is a set of adjacent pixels of pixel j. In case of a 2D problem the $N(j)$ consists of the pixel that is 1 pixel above the pixel j and of the pixel that is 1 pixel to the left from the pixel j. From our experience the performance of different norms do not vary much, and l_2 norm has some advantages in terms of flexibility of segmentation and it is significantly easier to optimize.

Taking into account (4), (9), (10), (11), the overall problem (3) has the following form:

$$\begin{aligned} \mathbf{x}^*, \delta = \arg\max_{\mathbf{x},\delta} \; & \lambda_1 \left[\sum_i b_i \log \sum_j a_{ij} x_j - \sum_i \sum_j a_{ij} x_j \right] + \\ & + \sum_j \log \sum_k \delta_{jk} \frac{1}{\sqrt{2\pi\sigma_k}} \exp\left(-\frac{(x_j - \mu_k)^2}{2\sigma_k^2}\right) - \\ & - \lambda_2 \Phi(\delta) \\ \text{s.t. } \forall j \quad & \sum_k \delta_{jk} = 1; \; \forall j,k \quad \delta_{jk} \geq 0 \\ & \forall j \quad x_j > 0 \end{aligned} \tag{13}$$

Here we introduce two regularization parameters λ_1, λ_2 that regulate how much the result is affected by the class fidelity and by neighborhood prior.

3 Simplification

The problem (13) is convex in terms of HMMFM δ, but is non-convex in terms of image $\mathbf{x}$. More than that, this problem for the majority of HMMFM realisations has many local minima. The main source of this non-convexity is the second term (10) of the optimization problem. To deal with this difficulty we make the substitution of the second term:

$$\tilde{p}(\mathbf{x}|\delta) = \prod_j \frac{1}{\sqrt{2\pi\tilde{\sigma}_j}} \exp\left(-\frac{(x_j - \tilde{\mu}_j)^2}{2\tilde{\sigma}_j}\right), \tag{14}$$

where the mean values and standard deviations are computed for each pixel:

$$\tilde{\mu}_j = \sum_k \delta_{j,k}\mu_k, \tag{15}$$

$$\tilde{\sigma}_j^2 = \sum_k \delta_{jk}(\sigma_k^2 + \mu_k^2) - \mu_j^2. \tag{16}$$

This simplification is convex in terms of $\mathbf{x}$, but non-convex in terms of δ. It is an approximation with a normal distribution of the distribution $p(\mathbf{x}|\delta)$ with the mean estimated with an expected value of this distribution and square of standard deviation estimated with variance of this distribution.

4 Algorithm

We minimize the problem iteratively and in the l-th iteration we would like to compute an approximation to the solution of the following problem:

$$\begin{aligned}
\mathbf{x}^{l+1}, \delta^{l+1} = \arg\max_{\mathbf{x},\delta} \; & \lambda_1 \left[\sum_i b_i \log \sum_j a_{ij}x_j - \sum_{i,j} x_j a_{ij}\right] + \\
& + \sum_j \log \sum_k \delta_{jk} \frac{1}{\sqrt{2\pi\sigma_k}} \exp\left(-\frac{(x_j - \mu_k)^2}{2\sigma_k^2}\right) - \\
& - \lambda_2 \Phi(\delta) \\
\text{s.t. } \forall j \quad & \sum_k \delta_{jk} = 1; \; \forall j,k \quad \delta_{jk} \geq 0 \\
& \forall j \quad x_j > 0.
\end{aligned} \tag{17}$$

Since to the problem is non-convex, we minimize the problem above using a two-stage algorithm. In the first stage we compute the approximation to the solution, while in the second stage we compute the final solution. In both stages we use the approximation to the second term of the problem to make the problem convex for the HMMFM.

4.1 First Stage

Since we have two variables to optimize, for one of which the problem is non-convex, we minimize this problem iteratively in two steps. In the first step we compute the approximation to the image $\mathbf{x}$ given the fixed HMMFM δ.

Thus, in the first step we would like to compute the solution to the part of the problem above that includes all the terms with the variable $\mathbf{x}$:

$$\begin{aligned}\mathbf{x}^{l+1} = \arg\max_{\mathbf{x}} \lambda_1 \left[\sum_i b_i \log \sum_j a_{ij} x_j - \sum_{i,j} x_j a_{ij} \right] + \\ + \sum_j \log \sum_k \delta_{jk}^l \frac{1}{\sqrt{2\pi\sigma_k}} \exp\left(-\frac{(x_j - \mu_k)^2}{2\sigma_k^2} \right) \\ \text{s.t. } \forall j \quad x_j > 0,\end{aligned} \tag{18}$$

but, as it was already mentioned, the second term here is non-convex. To deal with this non-convexity we use the approximation (14):

$$\begin{aligned}\mathbf{x}^{l+1} = \arg\max_{\mathbf{x}} \lambda_1 \left[\sum_i b_i \log \sum_j a_{ij} x_j - \sum_{i,j} x_j a_{ij} \right] + \\ + \sum_j \frac{(x_j - \tilde{\mu}_j^l)^2}{2(\tilde{\sigma}_j^l)^2} \\ \text{s.t. } \forall j \quad x_j > 0.\end{aligned} \tag{19}$$

With this approximation the problem becomes convex and we can easily minimize it using any optimization technique. We use a Limited Memory BFGS (L-BFGS) optimization algorithm [**?**] to minimize it. As we do many iterations of this two-steps algorithm - it is enough to do just few iterations of the L-BFGS - and we will get the next iteration of the image.

Although the problem (**??**) is convex, the overall problem is not. To deal with it we tried two approaches. One approach is to start with big data regularization λ_1 and gradually reduce it to a small value that was specified before the algorithm starts. Another approach is instead of λ_1 to gradually modify σ_k along the iterations.

We propose in the first approach to use the following expression for the regularization parameter:

$$\lambda_1^l = \lambda_1(1 + C\beta^l), \tag{20}$$

and for the second approach we propose to use the similar expression:

$$\sigma_k^l = \sigma_k(1 + C\beta^l), \tag{21}$$

where C and β is a constant. We require that $\lambda_1^l \to \lambda_1$ for $l \to \infty$ and $\sigma_k^l \to \sigma_k$ for $l \to \infty$. In order to achieve that, we should assign constant β to belong to the interval $(0, 1)$. In our experiments we use $C = 1000$, $\beta = 0.9$.

For the first approach the image reconstruction step will turn into the following problem:

$$\begin{aligned}\mathbf{x}^{l+1} = \arg\max_{\mathbf{x}} \lambda_1^l \left[\sum_i b_i \log \sum_j a_{ij} x_j - \sum_{i,j} x_j a_{ij} \right] + \\ + \sum_j \frac{(x_j - \tilde{\mu}_j^l)^2}{2(\tilde{\sigma}_j^l)^2} \\ \text{s.t. } \forall j \quad x_j > 0,\end{aligned} \tag{22}$$

for the second approach we need to recompute the values for $\tilde{\sigma}_j^l$ according to the rule 16, where σ_k should

be substituted by σ_k^l. Then the reconstruction step turns into the following problem:

$$\begin{aligned}\mathbf{x}^{l+1} = \arg\max_{\mathbf{x}} \lambda_1 \left[\sum_i b_i \log \sum_j a_{ij} x_j - \sum_{i,j} x_j a_{ij} \right] + \\ + \sum_j \frac{(x_j - \tilde{\mu}_j^l)^2}{2(\tilde{\sigma}_j^l)^2} \\ \text{s.t. } \forall j \quad x_j > 0.\end{aligned} \tag{23}$$

In principle, we could iterate L-BFGS to convergence to the solution of (23), but since this computation is only one step of our iterations, it is enough to improve the solution by few iterations of L-BFGS algorithm.

Once the image is updated we should find the next iteration of the segmentation. In the full problem (17) we keep the $\mathbf{x}$ fixed and optimize δ:

$$\begin{aligned}\delta^{l+1} = \arg\max_{\delta} \sum_j \log \sum_k \delta_{jk} \frac{1}{\sqrt{2\pi\sigma_k}} \exp\left(-\frac{(x_j^{l+1} - \mu_k)^2}{2\sigma_k^2} \right) - \\ - \lambda_2 \Phi(\delta) \\ \text{s.t. } \forall j \quad \sum_k \delta_{jk} = 1; \ \forall j,k \quad \delta_{jk} \geq 0.\end{aligned} \tag{24}$$

This is a convex problem with constraints. The constraints of this problem are simplices and we minimize it with a modified Frank-Wolfe algorithm that is specially designed to minimize the problems with these constraints.

Again, it is possible to do these iterations until convergence, but it is sufficient to make small improvement of the result.

4.2 Second Stage

In our original algorithm we also included a second stage. In this work we do not implement this approach, but we still want to formulate it. The purpose of the second stage is to force the solution to be closer to the class prior.

The second stage of the algorithm is very similar to the first stage: it consists of iterations that include the image optimization and HMMFM optimization using the same optimization techniques. The only difference here is the approximation of the second term of the problem (17).

In the previous stage we assumed that the distribution of the grey levels in the pixel could be represented by a single Gaussian with a mean value that corresponds to the mean value of the mixture of Gaussians and the standard deviation that corresponds to the standard deviation of the mixture of the Gaussians (15), (16). In this stage we assume that the grey levels distribution in each pixel is a Gaussian distribution with the mean value $\tilde{\mu}_j$ equal to the mean value of the most probable class and standard deviation equal to the standard deviation of the most probable class.

In other words, if

$$k_j = \arg\max_k \delta_{jk}$$

then

$$\tilde{\mu}_j = \mu_{k_j}, \tilde{\sigma}_j = \sigma_{k_j}.$$

As in the previous stage, first we do the image optimization

$$\begin{aligned}\mathbf{x}^{l+1} = \arg\max_{\mathbf{x}} \lambda_1 \left[\sum_i b_i \log \sum_j a_{ij} x_j - \sum_{i,j} x_j a_{ij} \right] + \\ + \sum_j \frac{(x_j - \tilde{\mu}_j)^2}{2\tilde{\sigma}_j^2}, \\ \text{s.t. } \forall j \quad x_j > 0.\end{aligned} \tag{25}$$

Require: $L_1, L_2, L_1 \leq L_2, \beta < 1, \lambda_1, \lambda_2, \{\mu_k\}, \{\sigma_k\}, A, \mathbf{b}$
  $\mathbf{x} \leftarrow$ ones
  $\delta \leftarrow$ equal probabilities
  $l \leftarrow 0$ (iteration index)
  while $l < L_1$ **do** ▷ SRS-II procedure, stage 1
    $l \leftarrow l + 1$
    Compute $\tilde{\mu_j}$, $\tilde{\sigma_j}$ according to (15), (16)
    Compute λ_1^l according to (20) or σ_k^l according to (21).
    $\mathbf{x}^{l+1} \leftarrow$ result of 20 iterations of L-BFGS optimization of (22) or (23) using $\tilde{\mu_j}$, $\tilde{\sigma_j}$ with fixed δ^l
    $\delta^{l+1} \leftarrow$ result of 20 iterations of Frank-Wolfe optimization of (24) with fixed $\mathbf{x}^{l+1}$
  end while
  while $l < L_2$ **do** ▷ SRS-II procedure, stage 2
    $l \leftarrow l + 1$
    $\mathbf{x}^{l+1} \leftarrow$ result of 20 iterations of L-BFGS optimization of (25) with fixed δ^l
    $\delta^{l+1} \leftarrow$ result of 20 iterations of Frank-Wolfe optimization of (26) with fixed $\mathbf{x}^{l+1}$
  end while
  return $\mathbf{x}^l$, δ^l

Figure 1: SRS-II algorithm.

And, knowing the new image approximation, we do the segmentation of this image:

$$\begin{aligned} \delta^{l+1} = \arg\max_{\delta} \sum_j \log \sum_k \delta_{jk} \frac{1}{\sqrt{2\pi\sigma_k}} \exp\left(-\frac{(x_j^{l+1} - \mu_k)^2}{2\sigma_k^2}\right) - \\ - \lambda_2 \|D\delta\|_2 \\ \text{s.t. } \forall j \quad \sum_k \delta_{jk} = 1; \ \forall j, k \quad \delta_{jk} \geq 0. \end{aligned} \tag{26}$$

After few iterations we can stop the process.

4.3 Additional aspects of the algorithm

As we have mentioned, the regularization parameter λ_1^l or the class standard deviations σ_k^l change from iteration to iteration according to formulae (20) (21). In our algorithm we use the following values in these formulae: $C = 1000$, $\beta = 0.9$. The plot of the values $\frac{\lambda_1^l}{\lambda_1}$ and $\frac{\sigma_k^l}{\sigma_k}$ can be found in Figure 10.

Quite important question is selection of the starting point for each of the the optimization subproblems in each iteration. We select the previous iterates for both image and HMMFM as a starting point for both optimization problems. As for the initial guess for the overall algorithm, we choose the initial image equal to all ones, the initial HMMFM is chosen as equal probabilities for all the classes in all the pixels, but the initial guess should not affect the result of the process.

The algorithm listing can be found on figure 1.

5 Computational Results

To check the reconstruction properties of our algorithm we have done a series of experiments. We generated artificial phantoms of size 384×384 pixels with following mean values: [33, 66, 99, 133], see Figure 2.

The geometry for the problem was generated using the package `ASTRA` [**?**]. The advantage of this package is that instead of generating the explicit projection matrix A which corresponds to the geometry of the scanning device, the information about the geometry is stored. That allows to make matrix-vector multiplications on the GPU.

We generated two data sets, one with 172 projections and the other with 86 projections, both sets have $\lfloor \sqrt{2} \cdot 384 \rfloor = 543$ rays in each projection. The projections were produced using parallel-beam geometry.

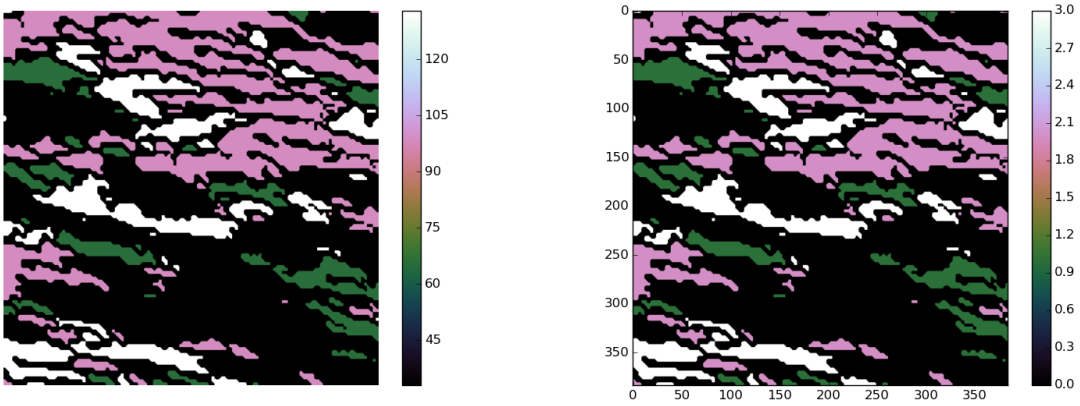

Figure 2: Ground truth of the experiment. Left - the phantom, right - the segmentation. The phantom is generated using the `AIR Tools` package.

The projection angles are evenly distributed between 0 and 180 degrees in both data sets. Thus, the ratio between amount of data and amount of pixels is $\frac{\#data}{\#pixels} = 0.62$ for the first dataset and $\frac{\#data}{\#pixels} = 0.31$ for the second dataset meaning that both problems are underdetermined. Moreover, some of the rays do not hit the phantom at all because of the square geometry of the phantom. We assume the noise in the data is Poisson-distributed, which corresponds to an Emission Tomography problem. Each measurement corresponds to the integral of the emission coefficient along a measurement line. We get the following quantity for the amount of noise:

$$\frac{\|\mathbf{b} - \mathbf{b}^*\|_2}{\|\mathbf{b}^*\|_2} = 0.0068,$$

for both data sets, where $\mathbf{b}^*$ is the data without noise.

The original phantom and the segmentation are shown in figure 2. The sinograms are shown in figure 3.

In the numerical experiments we expect to see that the use of the segmentation in the reconstruction process, jointly updating the image and the segmentation, will improve both the image and the segmentation. In the beginning of the iterations the regularization parameter λ_1^l is quite big meaning that the class fitting term and the regularization term have very little influence on the result of the optimization process. The fact that we get a better reconstruction during the iterations indicates that using the segmentation in the reconstruction process we are able to produce a better reconstruction. This is due to the "feedbck effect" between the segmentation and the reconstruction.

For the reconstruction process the following parameters were chosen: for the problem with 170 projections: $\lambda_1 = 250, \lambda_2 = 0.8$, for the problem with 85 projections: $\lambda_1 = 800, \lambda_2 = 1.0$. These parameters were obtained using trial-and-error and produced the best possible result.

We show the evolution of the reconstructions of the image and segmentations as well as the differences between the reconstruction of the phantom and misclassified pixels. Figures 4 and 5 show results for 170 projections and figures 6, 7 show results for 85 projections. The amount of misclassified pixels decreases with iterations, the quality of reconstruction increases over the iterations. We also see that with iterations the precision of location of the edges increases. From the figures one can conclude that the majority of misclassifications in the segmentation and errors in the reconstruction occur on the edges between the classes. This is easy to explain: the lower is the amount of photons - the less reliable is the data and the less precise the possible reconstruction could be. As a result, the position of the edge is less predictable, and this leads to the misclassifications on the edges. The error histories are presented in figure 8 for 170 projections and figure 9 for 85 projections. As a measure for the image error we used the relative l_1 error:

$$\frac{\|\mathbf{x}^l - \mathbf{x}^*\|_1}{\|\mathbf{x}^*\|_1},$$

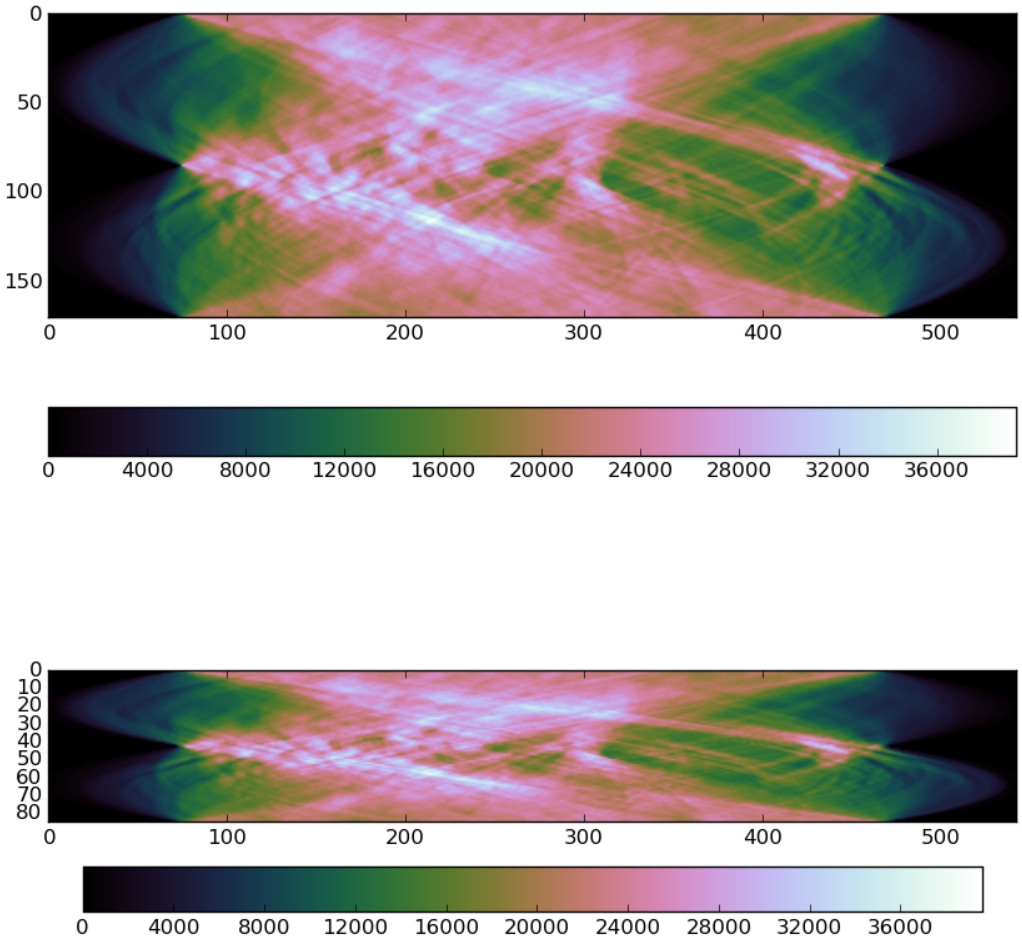

Figure 3: The sinograms of two experiments. Top: for 170 projections, bottom: for 85 projections. The noise level in the data is $\frac{\|\mathbf{b}-\mathbf{b}^*\|_2}{\|\mathbf{b}^*\|_2} = 0.0068$ in both cases, $\mathbf{b}^*$ is the data without noise, amount of rays in each of the projections is 543.

where $\mathbf{x}^*$ is the true image We have chosen this error because it penalizes small deviations and big deviations equally and, because of that, corresponds better to visual inspection. As an error measure for the segmentation we have chosen the fraction of misclassified pixels:

$$\frac{\sum_j I(s_j \neq s_j^*)}{\#pixels},$$

where I is a indicator function that is 0 when the argument is false and 1 when the argument is true; s^* is a true segmentation.

We can see that both reconstructions and segmentations improve along the iterations. In the same time, the reconstructions that were obtained with these methods are piecewise constant and the edges are sharp, staircasing artifacts are minimal. Besides, the noise is suppressed by the regularization.

We have also compared the performance of the different flavours of the algorithm (Table 2). First

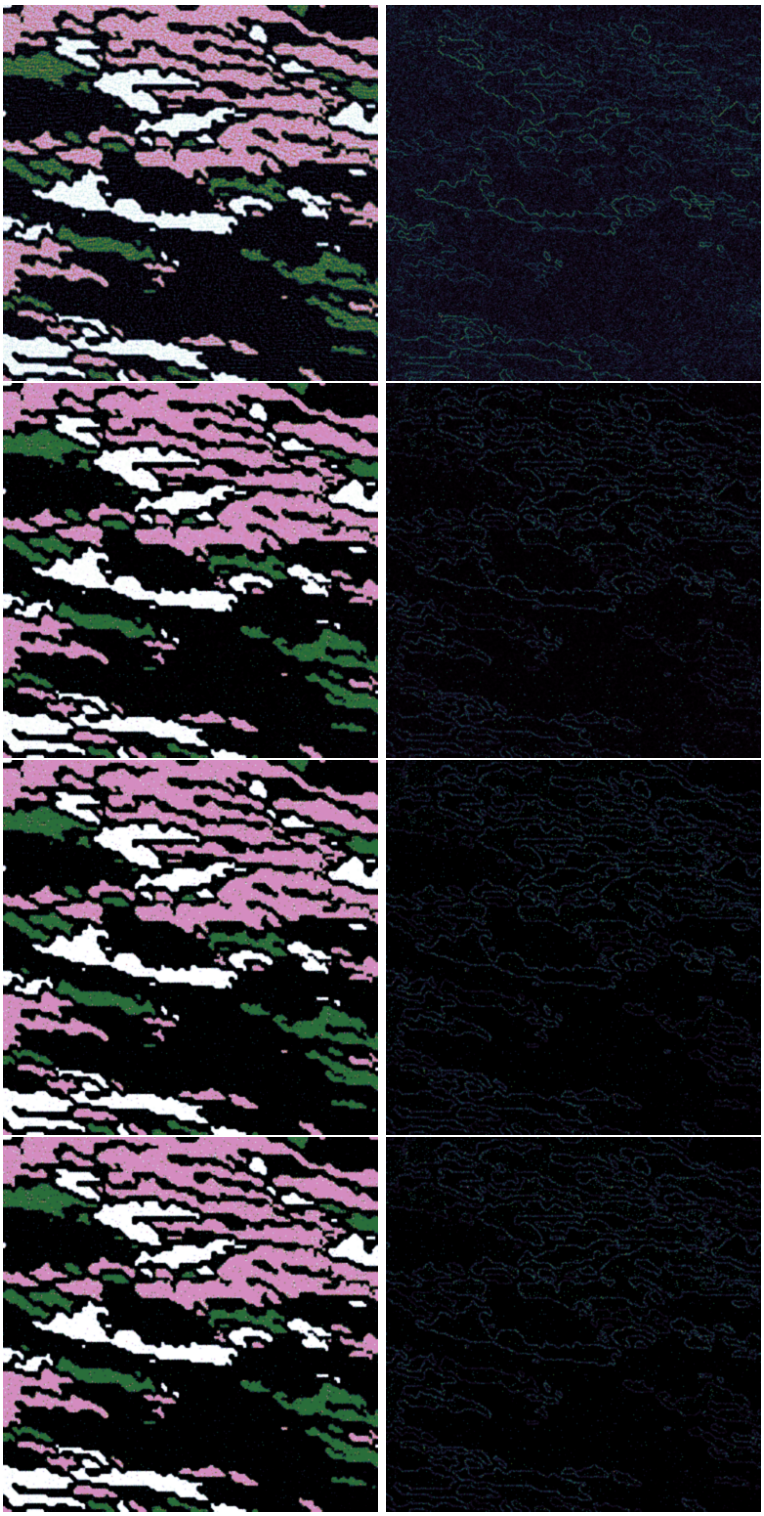

Figure 4: Left - reconstructed images, right - difference between current image and the ground truth $|\mathbf{x}^l - \mathbf{x}^*|$ (the $\mathbf{x}^*$ is a true image) for SRS-II algorithm applied to the test problem with 170 projections. The following iterations represented: 1, 33, 66, 100.

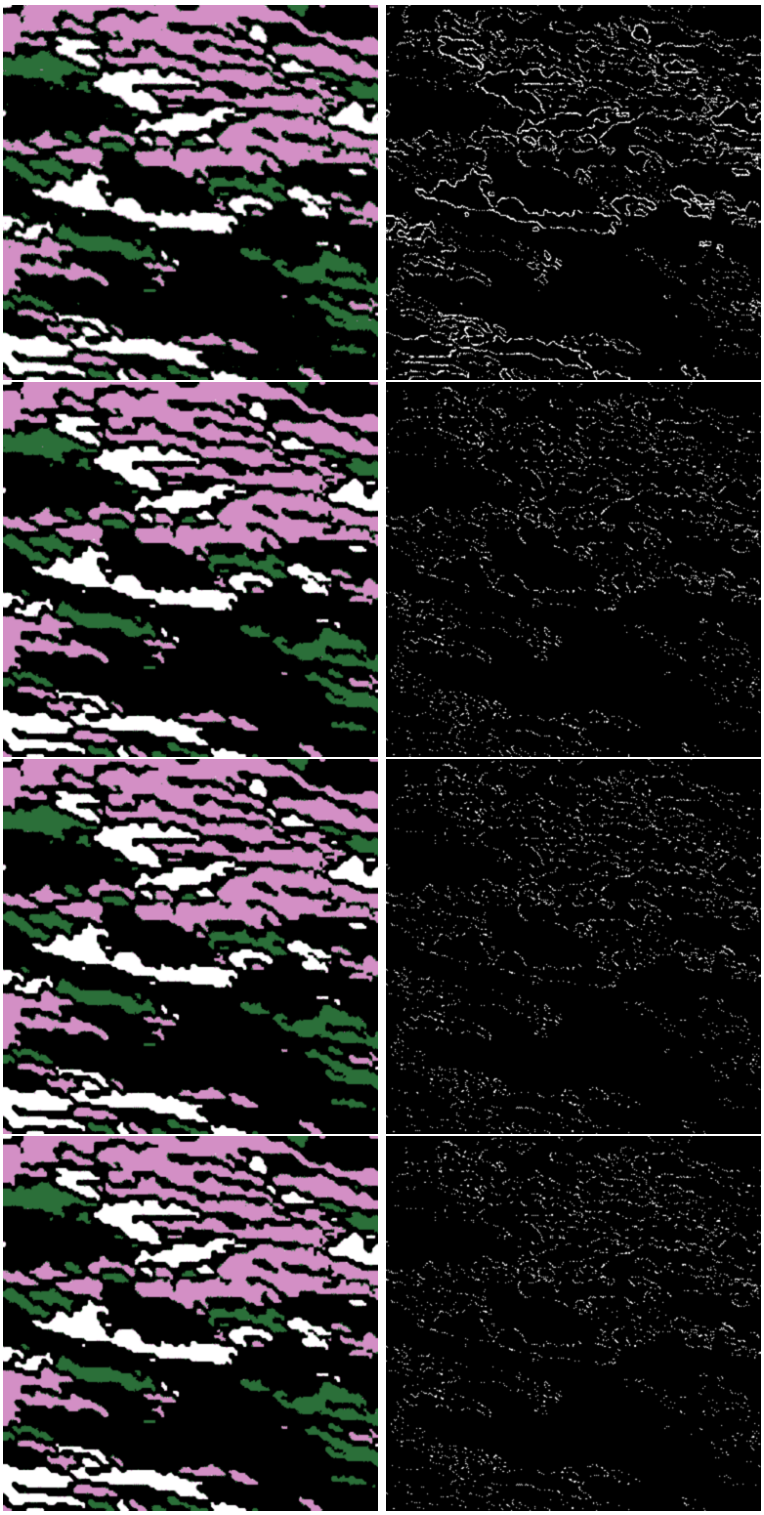

Figure 5: Left - computed segmentations, right - misclassified pixels (white) for SRS-II algorithm results computed from 170 projections. The following iterations are presented: 1, 33, 66, 100.

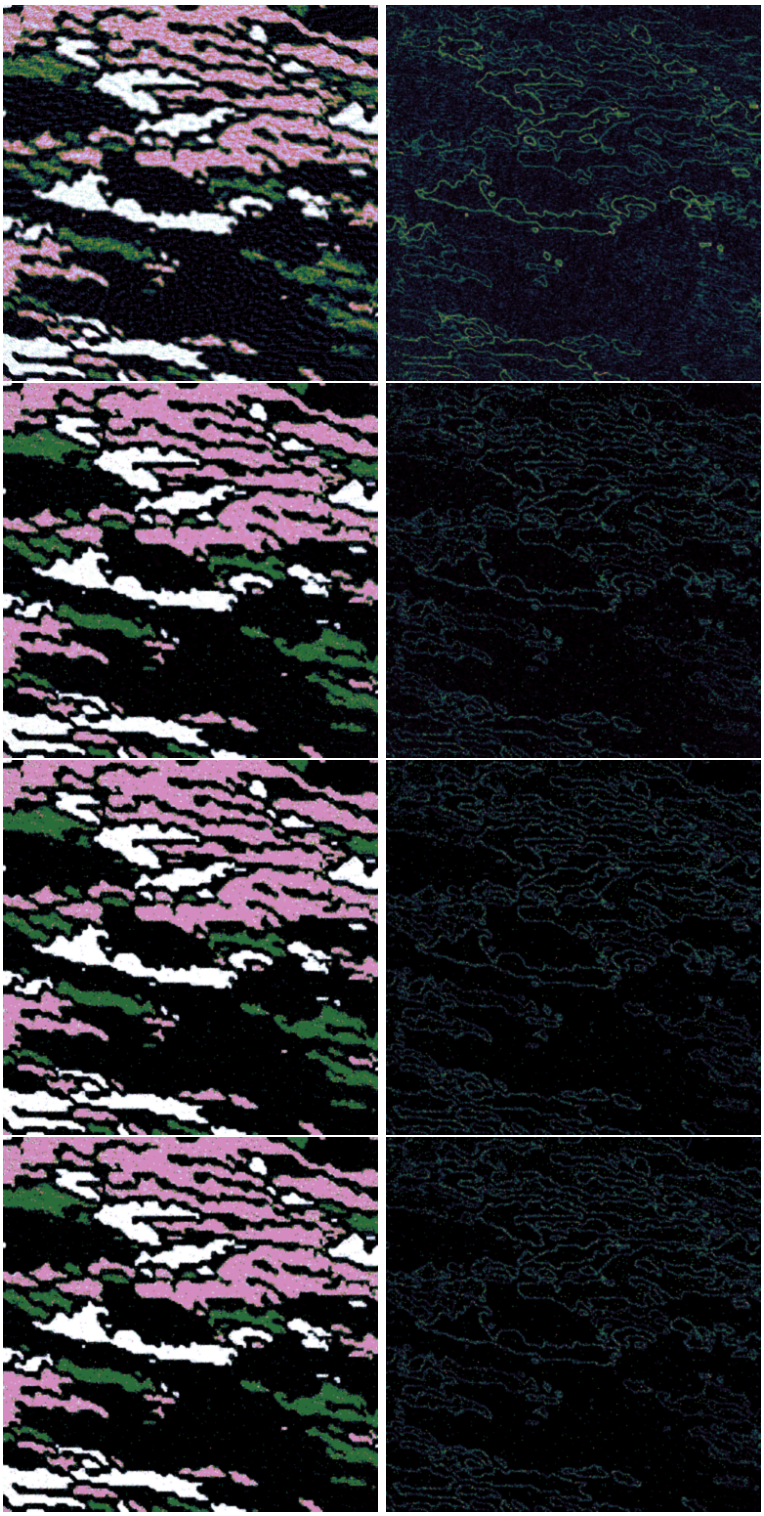

Figure 6: Left - reconstructed images, right - difference between current image and the ground truth $|\mathbf{x}^l - \mathbf{x}^*|$ (the $\mathbf{x}^*$ is a true image) for SRS-II algorithm applied to the test problem with 85 projections. The following iterations represented: 1, 33, 66, 100.

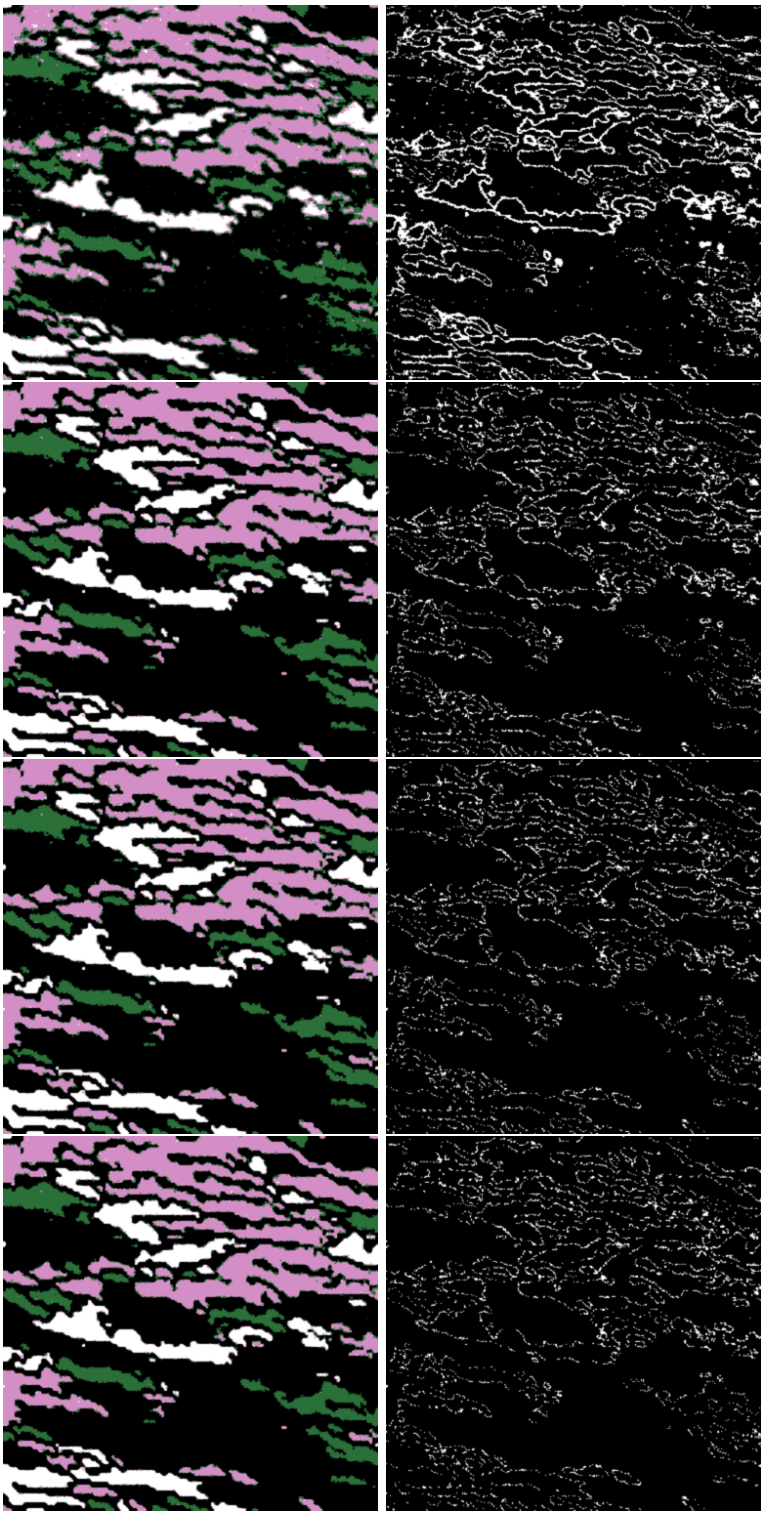

Figure 7: Left - computed segmentations, right - misclassified pixels (white) for SRS-II algorithm results computed from 85 projections. The following iterations are presented: 1, 33, 66, 100.

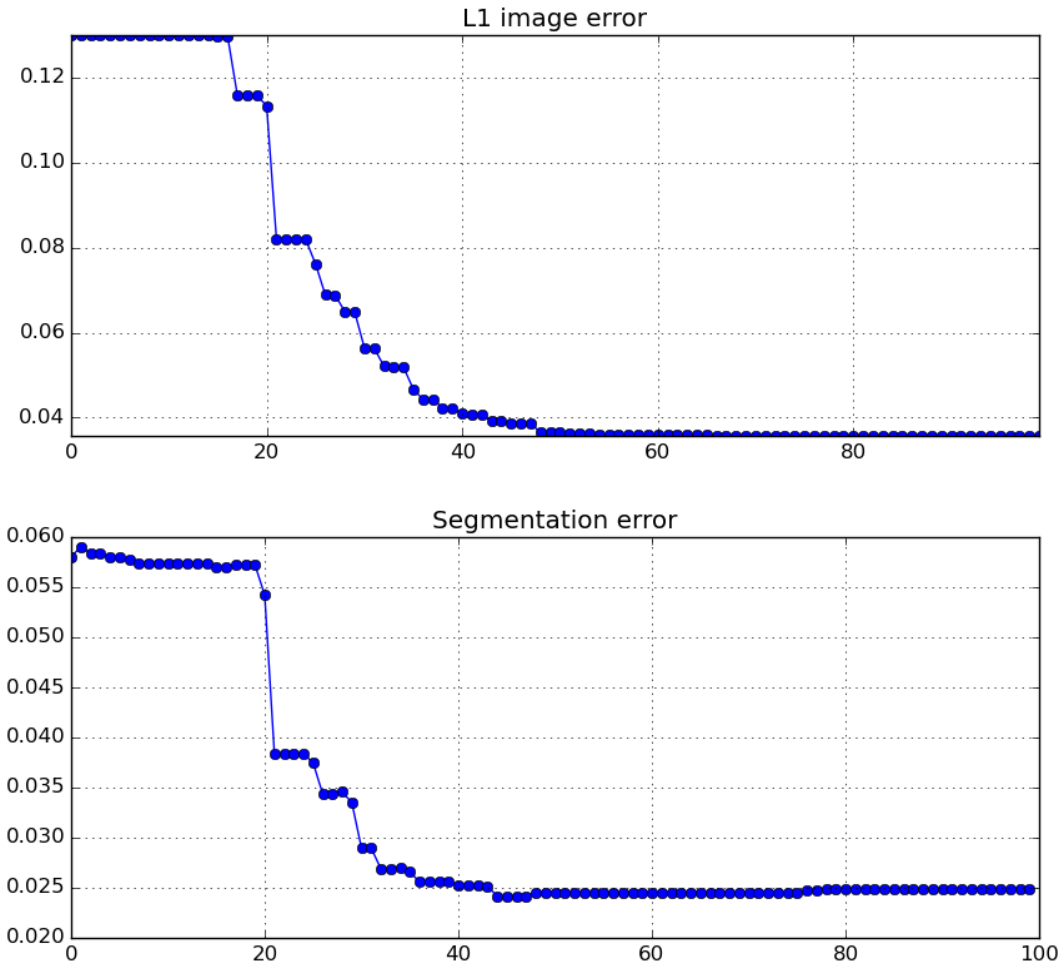

Figure 8: Error histories for SRS-II problem with 170 projections. Top: error history for l_1 image error $\frac{\|\mathbf{x}-\mathbf{x}^*\|_1}{\|\mathbf{x}^*\|_1}$. Bottom: segmentation error - the fraction of misclassified pixels. The $\mathbf{x}^*$ is a true image.

experiment was done with the SRS-II algorithm with 2-norm data fitting term

$$\|A\mathbf{x} - \mathbf{b}\|_2$$

with the data with $\frac{\#data}{\#pixels} = 0.3$ ratio between amount of data and amount of pixels (85 projections) and with 0.01 relative Gaussian noise. The reconstruction algorithm used standard deviations σ_k that were equal for all the classes and were identical and equal to 0.001. The regularization parameters that were used for this experiment are: $\lambda_1 = 0.9, \lambda_2 = 1.0$. Another experiment used Poisson data fitting term (the definition

		Gaussian $\sigma_k = 0.001$, $\frac{\#data}{\#pixels} = 0.3$	Poisson $\sigma_k = 0.001\mu_k$, $\frac{\#data}{\#pixels} = 0.3$	Poisson $\sigma_k = 0.001$, $\frac{\#data}{\#pixels} = 0.3$	Poisson $\sigma_k = 0.001$, $\frac{\#data}{\#pixels} = 0.6$
l_1 image error	σ_k reduction	0.06	0.07	0.061	0.035
	λ_1 reduction	-	0.077	0.08	0.048
segm. error	σ_k reduction	0.052	0.062	0.056	0.023
	λ_1 reduction	-	0.085	0.077	0.037

Table 2: The table of comparison of different reconstruction approaches.

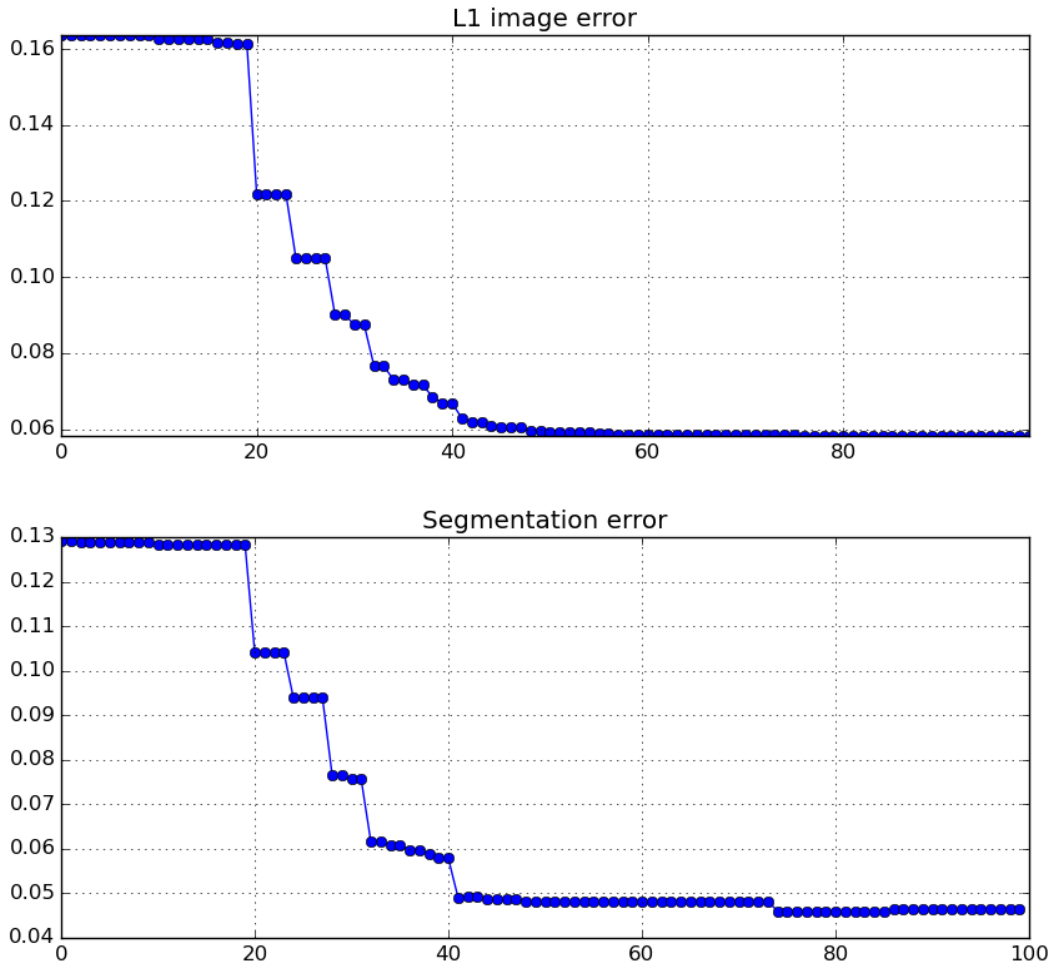

Figure 9: Error histories for SRS-II problem with 85 projections. Top: error history for l_1 image error errors $\frac{\|\mathbf{x}-\mathbf{x}^*\|_1}{\|x^*\|_1}$. Bottom: segmentation error - the fraction of misclassified pixels. The $\mathbf{x}^*$ is a true image.

for the data fitting term for the Poisson noise is given in (17)), the amount of data over the amount of pixels was $\frac{\#data}{\#pixels} = 0.3$ (85 projections), relative amount of Poisson noise was 0.0068, the standard deviations of classes were set to be equal to $\sigma_k = 0.001\mu_k$. The regularization parameters that were used for all the other problems are: $\lambda_1 = 2000.0, \lambda_2 = 0.8$. The difference between the third experiment and the second is only in the standard deviations of the classes. We need this comparison to be able to tell if the choice of the standard deviations may compensate for the features of the Poisson Noise: it tends to have bigger errors in places, where the intensity is higher. Here we used the standard deviations that are equal to 0.01. For comparison we added one more numerical experiment with more data: the ratio between amount of data and amount of pixels $\frac{\#data}{\#pixels} = 0.6$ (170 projections).

From this table one can make the following conclusion: though it may seem logical to use the standard deviation of the classes proportional to the mean values of the classes - this changes the results insignificantly and does not improve them. The problems with more data produce better results with the same regularization parameters. This is logical and predictable. Also, the algorithm is good for the reconstruction of the image from the data with Gaussian Noise too.

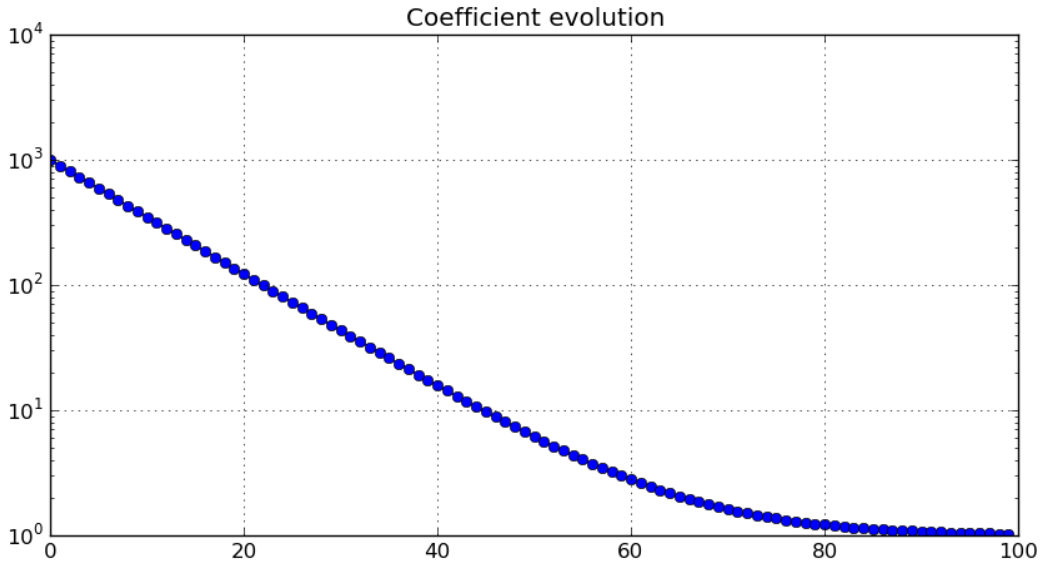

Figure 10: Evolution of the coefficients.

6 Conclusion

We have developed an algorithm for Poisson noise in the data. It is suited for large-scale-problems as we do not store the matrix A explicitly. The algorithm performs well on artificial test problems. The priors σ_k for the classes can be either fixed, or proportional to μ_k, but the computed results are almost the same. The solutions that are generated by this algorithm have sharp edges.

A Appendix: Explanation of Modified Standard Deviation

In the expression (23) and (22) we use the term σ_k^l and λ_1^l to gradually shrink the searching range of the image, starting with big standard deviations σ_k^l or high λ_1^l values that correspond to significantly underregularized problems. Along the iterations we reduce values λ_1^l and σ_k^l to the values λ_1 and σ_k and, by that we gradually increase the regularization of the problem.

To show the importance of this modification we will make the thought experiment: consider the problem (19), where $\tilde{\sigma}$ and $\tilde{\mu}$ are computed using the formulae (15), (16) two specific realization of δ. As an information about the classes we will take two classes

$$\mu_0 = 1,\ \mu_1 = 2,$$
$$\sigma_0 = 0.1,\ \sigma_1 = 0.1.$$

As an example of different HMMFM realizations we will take one with all equal probabilities:

$$\forall j, k\ \delta_{jk} = 1/2,$$

and another realization will have

$$\forall j\ \delta_{j0} = 1,\ \delta_{j1} = 0.$$

In both cases it is easy to estimate parameters $\tilde{\mu}_j$ and $\tilde{\sigma}_j$ for each pixels: in the first case the approximate parameters for each pixel are

$$\tilde{\mu}_j = 0.5,\ \tilde{\sigma}_j = \sqrt{0.26} \approx 0.5.$$

in the second case the approximate parameters for each pixel are:

$$\tilde{\mu}_j = 1.0,\ \tilde{\sigma}_j = 0.1.$$

In the first case - the standard deviation is large enough - meaning that any combination of grey levels of the pixels in the range between 0 and 1 is acceptable as far as it satisfies the data fitting term.

In the second case the standard deviation is small, meaning that the acceptable range of the grey levels of the pixels is in the range from 0.9 to 1.1. Hence, the data in this case may be fitted significantly worse due to these limits - and that means that the image that we can get in the end of the optimization process is overregularized.

In case after this image optimization step we will try to optimize the HMMFM - in the first case we will get HMMFM that corresponds to the image that fits data quite well. In the second case - we will get a HMMFM that corresponds to the overregularized image - meaning that HMMFM has very small chance to change.

That automatically means that different initializations of the HMMFM may lead to completely different results due to this fact. Another problem that is a consequence of this effect is that the algorithm without modifications tends to stuck in the local minimum. We would like to get rid of both these problem that may spoil the results of the algorithm.

Consider now the modified standard deviations:

$$\sigma_k^l = \sigma_k(1 + C\beta^l).$$

Consider also $\sigma_k = 0.001$, $C = 1000$, $\beta = 0.9$. In the second case the result of the 15 and 16 with $l = 0$ will be

$$\tilde{\mu}_j = 1.0, \tilde{\sigma}_j = 1.0$$

that will mean that the image in the beginning (when l is small) can change significantly. On the other hand, consider $l = 100$. In this case the value of the expression 15 will not change, but the value of 16 will be

$$\sigma_j \approx 0.001$$

meaning that in the end the result is regularized significantly by HMMFM δ. This approach is related to the Simulated Annealing approach.

The same effect will take place when, instead of σ_k we shrink the λ_1 over the iterations:

$$\lambda_1^l = \lambda_1(1 + C\beta^l).$$

Although, in the first case the form of the class prior also changes, while in this case only the level of regularization changes. From our experience, it is better to change σ_k over the iterations, but shrinking λ_1 it is also possible to get a good result.

References

[1] Dominique Van de Sompel and Michael Brady. Simultaneous reconstruction and segmentation algorithm for positron emission tomography and transmission tomography. In *Proceedings of the 5th IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, pages 1035–1038, 2008.

[2] Mikhail Romanov, Anders Bjorholm Dahl, Yiqiu Dong, and Per Christian Hansen. Simultaneous toographic reconstruction and segmentation with class priors. *subitted to Inverse Problems in Science and Engineering*, 2015.

[3] Ronald Newbold Bracewell and ACf Riddle. Inversion of fan-beam scans in radio astronomy. *The Astrophysical Journal*, 150:427, 1967.

[4] Avinash C. Kak and Malcolm Slaney. *Principles of Computerized Tomographic Imaging*. SIAM, Philadelphia, 2001.

[5] Per Christian Hansen and Maria Saxild-Hansen. AIR Tools – a MATLAB package of algebraic iterative reconstruction methods. *J. Comput. Appl. Math.*, 236(8):2167–2178, 2012.

[6] S Kaczmarz. Approximate solution of systems of linear equations. *International Journal of Control*, 57(6):1269–1271, 1993.

[7] Carl D Meyer. *Matrix analysis and applied linear algebra*. SIAM, 2000.

[8] Tommy Elfving, Per Christian Hansen, and Touraj Nikazad. Semi-convergence properties of kaczmarz's method. *Inverse Problems*, 30(5):055007, 2014.

[9] Junguo Bian, Jeffrey H Siewerdsen, Xiao Han, Emil Y Sidky, Jerry L Prince, Charles A Pelizzari, and Xiaochuan Pan. Evaluation of sparse-view reconstruction from flat-panel-detector cone-beam CT. *Physics in Medicine and Biology*, 55(22):6575, 2010.

[10] Emil Y Sidky, Chien-Min Kao, and Xiaochuan Pan. Accurate image reconstruction from few-views and limited-angle data in divergent-beam CT. *Journal of X-ray Science and Technology*, 14(2):119–139, 2006.

[11] Tony F Chan, Gene H Golub, and Pep Mulet. A nonlinear primal-dual method for total variation-based image restoration. *SIAM Journal on Scientific Computing*, 20(6):1964–1977, 1999.

[12] David Strong and Tony Chan. Edge-preserving and scale-dependent properties of total variation regularization. *Inverse Problems*, 19(6):S165, 2003.

[13] Michael Kass, Andrew Witkin, and Demetri Terzopoulos. Snakes: Active contour models. *Int. J. Comp. Vis.*, 1(4):321–331, 1988.

[14] Yuri Boykov, Olga Veksler, and Ramin Zabih. Fast approximate energy minimization via graph cuts. *IEEE Trans. Pattern Anal. Mach. Intell.*, 23(11):1222–1239, 2001.

[15] Yuri Boykov and Gareth Funka-Lea. Graph cuts and efficient ND image segmentation. *Int. J. Comp. Vis.*, 70(2):109–131, 2006.

[16] Yuri Y Boykov and M-P Jolly. Interactive graph cuts for optimal boundary and region segmentation of objects in ND images. In *Proceedings of the Eighth IEEE International Conference on Computer Vision*, volume 1, pages 105–112. IEEE, 2001.

[17] James Albert Sethian. *Level Set Methods and Fast Marching Methods: Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision, and Materials Science*. Cambridge University Press, 1999.

[18] Stanley Osher and Nikos Paragios. *Geometric Level Set Methods in Imaging, Vision, and Graphics*. Springer, 2003.

[19] Kyongtae T Bae, Maryellen L Giger, Chin-Tu Chen, and Charles E Kahn Jr. Automatic segmentation of liver structure in CT images. *Medical Physics*, 20(1):71–78, 1993.

[20] Jose L Marroquin, Edgar Arce Santana, and Salvador Botello. Hidden Markov measure field models for image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 25(11):1380–1387, 2003.

[21] Richard H Byrd, Peihuang Lu, Jorge Nocedal, and Ciyou Zhu. A limited memory algorithm for bound constrained optimization. *SIAM Journal on Scientific Computing*, 16(5):1190–1208, 1995.

[22] Dimitri P. Bertsekas. *Nonlinear Programming, 2. Ed.* Athena Scientific, Belmont, MA, 1999.

[23] Willem Jan Palenstijn, K Joost Batenburg, and Jan Sijbers. The astra tomography toolbox. In *13th International Conference on Computational and Mathematical Methods in Science and Engineering, CMMSE*, volume 2013, 2013.